Decomposable MAC Framework for Highly Flexible and Adaptable MAC Realizations

Junaid Ansari, Xi Zhang, Andreas Achtzehn, Marina Petrova and Petri Mähönen
Institute for Networked Systems, RWTH Aachen University, Kackertstrasse 9, D-52072, Aachen, Germany
Email: {jan, xzh, aac, mpe, pma}@mobnets.rwth-aachen.de

***Abstract*— Cognitive radios are becoming reality also in implementation domain. Besides the need for hardware reconfigurability and the capability to sense spectrum opportunities, adaptability in the MAC designs is required so that the wireless communication systems can support cognitive radio functionalities. In this demo paper, we introduce a MAC design framework enabling fast composition of MAC protocols which are best fitted to the application requirements, communication capabilities of the radio, and current regulations and policies. Our design is based on the decomposition principle and allows on-the-fly realization of the required MAC protocol from a set of basic functional components. By exposing extended metadata and hardware functionalities for the MAC implementation through our granular components, together with the support for run-time re-configuration, spectrum agile and cognitive MAC solutions can be easily realized. We validate our approach through realization of a few MAC solutions on the WARP board originally from Rice University, USA. We also demonstrate the ease of MAC realization, fast real-time adaptation based on the spectral characteristics and high degree of code reuse.**

I. Introduction

Cognitive radios are becoming an enabling technology for efficiently managing the constrained spectral resources and fulfilling varying degree of QoS demands. Intelligent management of spectral resources and advanced sensing in medium access procedures require high degree of adaptability and close interaction between the PHY/MAC functionalities. Although efforts have been made by the research community towards providing re-configurable and dynamic solutions for MAC realizations, the research still lacks a flexible framework for rapid on-the-fly adaptation and access/control to fine-grained radio/hardware functionalities. One of the practical shortcomings in MAC development for dynamic spectrum access has been that many platforms and their interfaces have restricted accessibility (e.g. IEEE 802.11 NICs). On the other hand, some of the more open SDR platforms such as WARP and USRP boards provide only limited MAC functionalities. This fact substantially curtails the development and experimental room for cognitive MACs and networks. Most of the MAC protocols are implemented in a monolithic fashion with tight coupling to the underlying hardware. This restricts adaptation and flexibility aspects required by spectrum agile and cognitive MACs. In order to address the issue of re-configuration, a few modular design approaches [1] have been proposed but they either lack actual implementation or are incapable to meet the real time requirements [2]. One of the major reasons for their shortcomings is the pure software implementation, which is unable to meet time-critical requirements as discussed in [3]. Unlike the earlier multi-MAC approaches promising reconfiguration aspects (e.g. [4]) which allows switching on a few pre-defined standalone MACs, our framework allows realizing a wide range of MAC implementations based on fundamental and elementary building blocks. We define a set of fundamental MAC functionalities as a library so that a wide range of MAC protocols can easily be realized by simply combining these functionalities in an appropriate manner. A re-wiring engine is designed in order to bind individual MAC functional blocks together and to coordinate the data/control flow among the blocks for rapid run-time MAC realizations. Profiling MAC implementations based on the fundamental blocks also indicates the key atomic blocks/operations requiring hardware acceleration, strict timing deadlines and high communication burden. Consideration of these factors certainly improves the implementation. The fundamental blocks expose wider hardware functionalities and leads to more access/control to the PHY-MAC parameters. This facilitates the design of cognitive MACs demanding higher degree of PHY-MAC interaction [5].

II. Design and Implementation of the Decomposable MAC Framework

A close interaction with the hardware interfaces and run-time re-configuration are needed by cognitive radio developers.

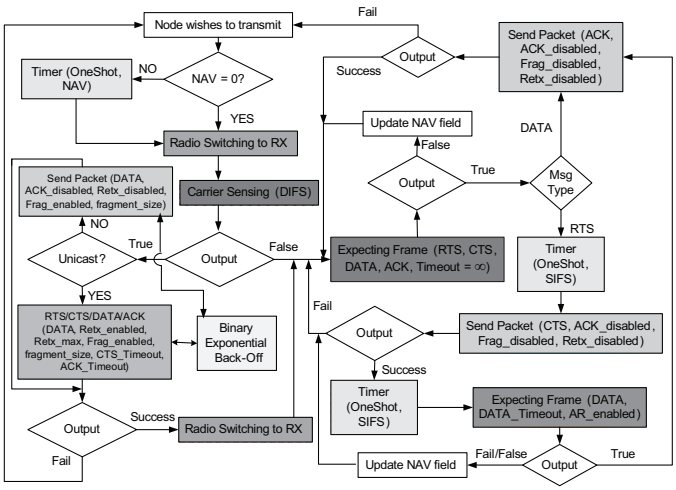

Fig. 1. Realization of IEEE 802.11 DCF using the elementary MAC blocks.

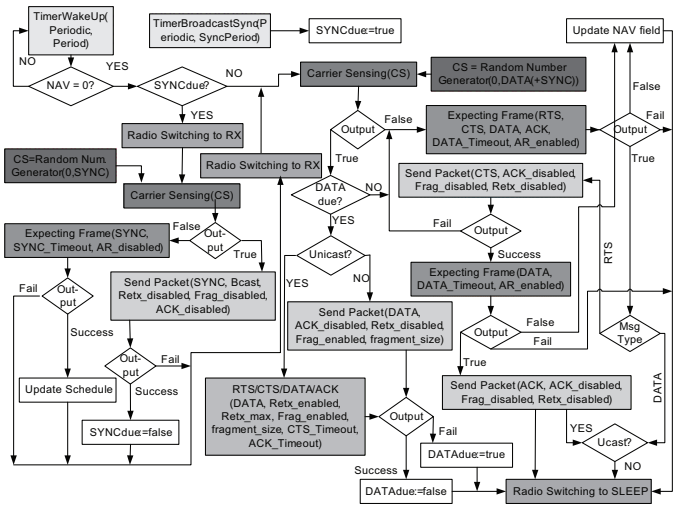

Fig. 2. Realization of S-MAC using the elementary MAC blocks.

Our framework helps the CR-community in this regard by providing extended hardware access/control interfaces and enabling re-configuration through a re-wiring engine. We have analyzed a wide range of MAC design approaches based on CSMA, TDMA and hybrid principles and identify the commonalities among different solutions. The commonalities among the MACs are based on their functional nature and their implementation specifics, e.g. timings, computational complexity and the degree of component reuse in a particular realization as well as across different protocols. With our approach, the actual implementation is merely reduced to the control/dataflow definitions using the elementary MAC blocks. Fig. 1 and Fig. 2 show the realization of IEEE 802.11 DCF and S-MAC, respectively. These figures also illustrate the concept of MAC realizations using the same set of gray scale coded fundamental blocks connected appropriately.

The elementary blocks in the framework are exposed through well-defined flexible APIs (Application Programming Interfaces), which in our demonstrator are implemented for the WARP board. The wiring-engine is a mechanism in our framework which is used to connect basic blocks according to the defined MAC functionalities. The framework is designed to be light-weight, enable code re-usage and minimize protocol implementation effort. Compositional adaptation techniques have been well investigated in Computer Science [6]. Our wiring-engine is designed based on the usage of function pointers. The construction and execute path of a state-machine is dynamically redirected through modification of function pointer assignments. It enables both run-time reconfiguration of MAC protocols according to pre-defined rules within the framework and on-the-fly realization of user configured protocols. For example, `CarrierSense()` is registered as the callback function for a periodic timer in a duty cycle based CSMA protocol. When the application requirements change and the framework has decided to switch to TDMA based protocol, the pointer to the callback function will be assigned to `SlotAssignment()` for protocol transition according to the already defined rules. This way hardware resources can efficiently be reallocated to the new tasks. When the user wishes to use a function which does not exist in the library, she can define and add the function to the library using `addFunction()` in a configuration file.

III. Demonstration Description and Conclusions

In the demonstration, we will present the architecture of the decomposable MAC framework and the identified MAC elementary blocks exposed through flexible APIs. We will demonstrate the ease of realizing a MAC protocol on the WARP board using the framework and our tool-chain. We base our implementation on the OFDM reference design v.14 based PHY for the WARP boards [7]. By expanding the original reference design of the WARP board with interrupt signalling capabilities, we are able to improve performance by minimizing the event-response delays. The MAC adapts itself on-the-fly depending on the changing spectral characteristics. User controlled interference generated through a WLAN access point (representing a primary user) will trigger the MAC to switch channel and reconfigure itself at the run-time. The reconfiguration includes DSA features with advanced carrier sensing, usage of additional control frames and adjustments of the backoff and persistency values in the protocol. Inclusion/exclusion of the blocks and appropriate re-wirings are required for this purpose. We show that our framework is capable of performing rapid runtime reconfigurations and bears high degree of code reuse in realizing different MAC protocols. Our framework will provide the cognitive radio development community a flexible tool for rapid development MAC layers in conjunction with other CR implementations once it is released in the future.

Acknowledgment

We would like to thank the financial support from RWTH Aachen University, DFG through UMIC research center and EU through 2PARMA project.

References

[1] D. G. Messerschmitt, "Rethinking components: From hardware and software to systems," *Proceedings of IEEE*, vol. 95, no. 7, pp. 1473–1496, 2007.

[2] A. Farago, A. Myers, V. Syrotiuk, and G. Zaruba, "Meta-MAC protocols: Automatic combination of MAC protocols to optimize performance for unknown conditions," *IEEE Journal on Selected Areas in Communications,*, vol. 18, no. 9, pp. 1670–1681, September 2000.

[3] G. Nychis, T. Hottelier, Z. Yang, S. Seshan and P. Steenkiste, "Enabling MAC protocol implementations on software-defined radios," in *Proceedings of the 6th USENIX symposium on Networked systems design and implementation (NSDI)*, April 2009, pp. 91–105.

[4] C. Doerr, M. Neufeld, J. Fifield, T. Weingart, D. Sicker, and D. Grunwald, "Multimac - an adaptive mac framework for dynamic radio networking," in *Proc. of DySPAN, Baltimore, MD, USA*, November 2005, pp. 548–555.

[5] C. Cormio and K. Chowdhury, "A survey on MAC protocols for cognitive radio networks," *Ad Hoc Netw.*, vol. 7, no. 7, pp. 1315–29, 2009.

[6] P. K. Mckinley, S. M. Sadjadi, E. P. Kasten and B. H. C. Cheng, "A Taxonomy of Compositional Adaption," Michigan State University, MSU-CSE-04-17, Tech. Rep., May 2004.

[7] "WARP Board OFDM reference design," http://warp.rice.edu/trac/wiki/OFDMReferenceDesign [Last visited: February 1, 2010].